

# Modul Pelatihan Teaching Factory Programming: Navigasi dan Lokalisasi dengan RTAB-Map

Program pelatihan ini dirancang untuk mahasiswa yang sudah memiliki pengetahuan dasar tentang ROS2. Fokus pelatihan adalah pada navigasi, lokalisasi, navigasi waypoint, dan pengendalian misi menggunakan state machine. Pelatihan ini menggunakan simulasi TurtleBot dan RTAB-Map.

## 1 Modul 1: Setup & Instalasi

### 1.1 Konfigurasi Environment

- Install ROS 2 Humble lakukan instalasi menggunakan **panduan instalasi Debian**
- Install TurtleBot3 packages menggunakan **panduan instalasi TurtleBot3**
- Install Gazebo dengan `sudo apt install ros-humble-desktop`
- Install RViz2 dengan `sudo apt install ros-humble-rviz2`
- Install RTAB-Map dengan `sudo apt install ros-humble-rtabmap-ros`
- Install Navigation2 dengan `sudo apt install ros-humble-navigation2 ros-humble-nav2-bringup`

### 1.2 Menjalankan Simulasi

Setelah semua paket terinstal, jalankan simulasi TurtleBot3 Waffle Pi di Gazebo dengan environment house:

```
export TURTLEBOT3_MODEL=waffle_pi
ros2 launch turtlebot3_gazebo turtlebot3_house.launch.py
```

### 1.3 Teleoperation

Buka terminal lain untuk menjalankan teleop:

```
ros2 run turtlebot3_teleop teleop_keyboard
```

### 1.4 RViz2

Buka RViz2 untuk melihat topik yang dipublish dan TF yang dimiliki robot:

```
rviz2
```

Atur Fixed Frame ke `odom`. Tambahkan display untuk `LaserScan`, dan TF. Sehingga akan muncul tampilan seperti berikut:

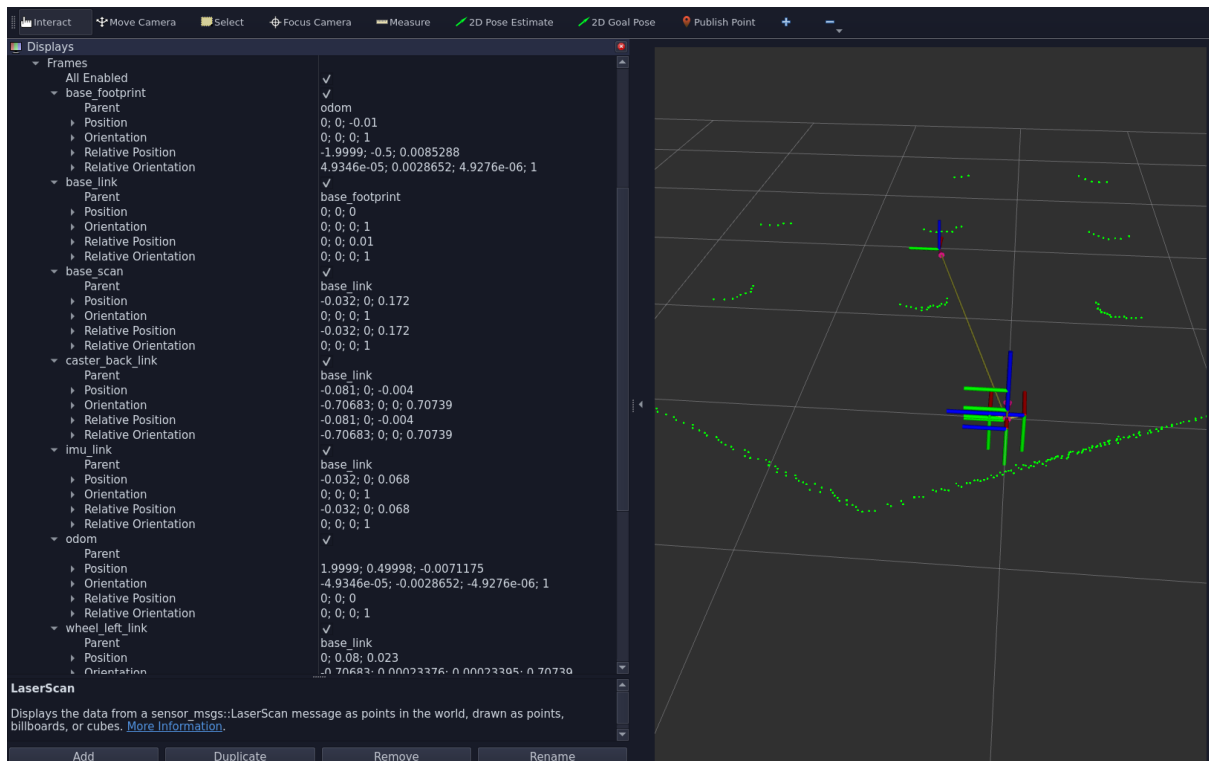


Figure 1: Contoh tampilan RViz2

Pada sidebar ditunjukkan TF yang dimiliki robot:

- **odom**: frame odometri, bergerak relatif terhadap **base\_link**
- **base\_footprint**: frame footprint robot (sejajar dengan ground plane 2D)
- **base\_link**: frame utama robot (dapat mengalami translasi dan orientasi)
- **caster\_back\_link**: frame caster back (free wheel)
- **imu\_link**: frame IMU
- **base\_scan**: frame laser scan
- **wheel\_left\_link**: frame roda kiri
- **wheel\_right\_link**: frame roda kanan

Semua TF tersebut terhubung dalam sebuah pohon (tree) yang dapat dilihat pada tab TF di RViz2. atau dengan perintah:

```
ros2 run tf2_tools view_frames.py
```

Perintah ini akan menghasilkan file **frames.pdf** yang menunjukkan struktur pohon TF sebagai berikut:

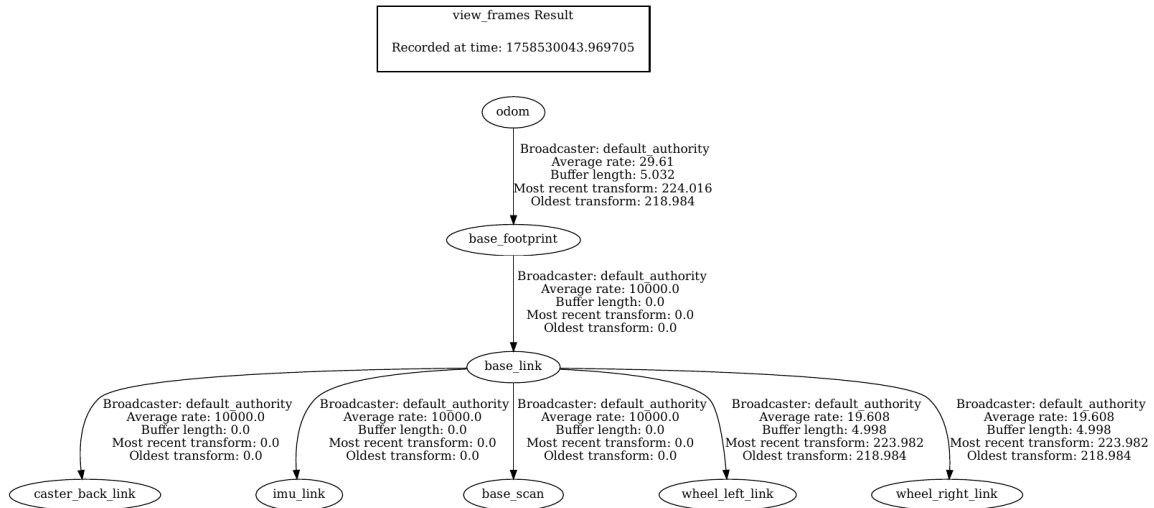


Figure 2: Contoh pohon TF

## 2 Modul 2: Mapping dengan RTAB-Map

### Objectives:

- Understand SLAM and RTAB-Map
- Run RTAB-Map in simulation
- Save and load maps

### Exercises:

- Launch TurtleBot3 with RTAB-Map
- Explore environment to build map
- Save map using `ros2 run nav2_map_server map_saver_cli`

## 3 Module 3: Localization with RTAB-Map

### Objectives:

- Use pre-built maps for localization
- Understand TF tree: `map` → `odom` → `base_link`

### Exercises:

- Launch localization node with `map.yaml`
- Move robot and verify pose in RViz

## 4 Module 4: Navigation with Nav2

### Objectives:

- Learn Navigation2 stack
- Configure global and local planners

- Tune costmap parameters

**Exercises:**

- Launch nav2.bringup with TurtleBot
- Send a navigation goal in RViz
- Observe path planning and execution

## 5 Module 5: Waypoint Navigation

**Objectives:**

- Define and manage multiple waypoints
- Create patrol behavior using waypoints

**Exercises:**

- Write Python node to send multiple goals
- Test waypoint navigation in simulation

## 6 Module 6: State Machines for Mission Control

**Objectives:**

- Understand state machines in robotics
- Implement mission logic with SMACH or FlexBE

**Exercises:**

- Create simple state machine with two states
- Expand mission with patrol → detect → return

## 7 Module 7: Advanced Topics & Real Robot Integration

**Objectives:**

- Handle kidnapped robot problem
- Fuse IMU + odometry + RTAB-Map
- Deploy to a real TurtleBot

**Exercises:**

- Test re-localization after manual robot displacement
- Run mapping on physical TurtleBot

## Final Project

Students implement a complete mission-based program:

- Explore and build map with RTAB-Map
- Localize using saved map
- Patrol using waypoints
- Implement mission logic with state machine